is now

indie Semiconductor FFO GmbH

To learn more about indie Semiconductor, please visit our website at www.indiesemi.com

For customer support, please contact us at: dfo.support@indiesemi.com

Silicon Radar GmbH
Im Technologiepark 1
15236 Frankfurt (Oder)
Germany

fon +49 (0) 335 228 80 30
fax +49.335.557 10 50
http://www.siliconradar.com

Protocol Description
SiRad Easy® - CW Mode

Status:	Date:	Author:	
Release	12 Sep. 19	Silicon Radar GmbH	
Version:	Document number:	Filename:	Page:
1.0.1		Protocol Description	1 of 18

Version Control

Version	Changed section	Description of change	Reason of change
1.0	all		Initial document
1.0.1	all	Extended mode description	Protocol update

Table of Content

List of Figures

1 Measurement Flow

This section describes the measurement flow of the SiRad Easy® in CW mode. The measurement parameters as well as the kind and amount of transmitted data can be set up using communication protocol described in the following sections.

Each measurement cycle is initiated by either an internal 'Self-Trigger' or an external 'Manual Trigger'. Continuous measurements can be triggered with a certain trigger frequency. The AD converter begins processing the chosen number of data samples with a certain sample frequency. All data is always transferred immediately after a measurement took place. The baseband amplifier may need to be modified depending on the application.

2 Standard Data

The SiRad Easy® CW mode communicates via UART. The baud rate is 1Mbaud, please make sure that you select the correct baud rate while connecting the SiRad Easy®.

The UART protocol supports comma separated decimal raw ADC values, and a binary format. The Silicon Radar WebGUI is not supported in CW mode. Therefore, command frames can only be sent from a terminal program.

There are several frame types that consist of multiple data values tied together to form a specific data packet. Each frame type is recognized by a unique identifier (a certain letter, 1 byte) and ends with a stop marker 'CR' + 'LF' (2 byte) as shown in Figure 1.

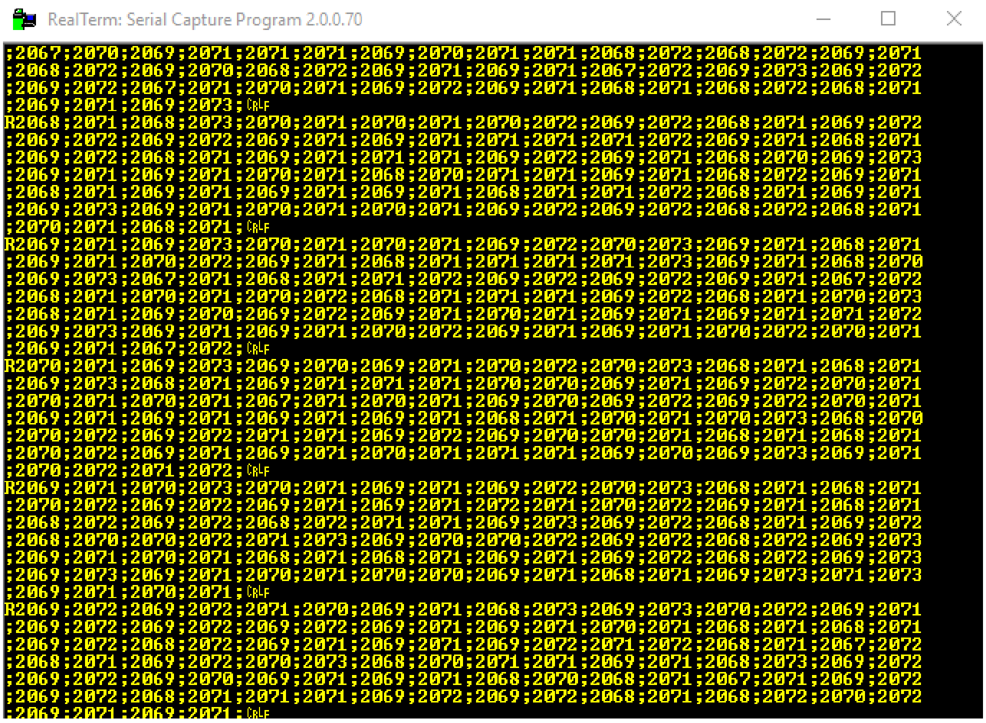

Figure 1: Raw data output in decimal format

Table 1 shows the available standard data frames.

Table 1: Standard data frames overview, description

Standard data frame	Description
Status update frame	Contains status data updates
System info frame	Contains hardware information
Error info frame	Contains error information
Version info frame	Contains hardware and firmware information
Detailed error info frame	Contains detailed error information

2.1 Status Information

The status update frame in Figure 2 is a feedback of the gain. The status update frame begins with the start marker (1 byte) and the identifier ‘U’ (1 byte) followed ‘Gain’ value (1 byte) for the last measurement. The frame ends with the stop markers ‘CR’ + ‘LF’.

Status information	Start	Identifier	Gain	Stop	
Status update frame	!	U	C	CR	LF

Figure 2: Status update frame format

Table 2: Status update frame - data encoding and interpretation of Gain

Field	Ecoding	Example	Interpretation	Allowed values
Gain	c - character between decimal value 34 and 254	letter ‘Z’ -> decimal 90	-140 to +80 dB in 220 steps	182, 195, 217, 230

Gain field

The ‘Gain’ field is transmitted as a character (marked with a ‘C’ in Figure 2). The data is recognized as a character of decimal value 34 to 254 in the terminal output, and is interpreted as a value between -140 and +80 dB in 220 steps of 1 dB. For example, ‘Gain’ = ’Z’ is decimal 90 and means -84 dB. There are currently four fixed gain settings available that depend on the hardware version.

2.2 Version Information

The version frame is used to uniquely identify the SiRad Evaluation Kit and returns information about the hardware and firmware, see Figure 3.

Version information	Start	Identifier	Length	UID tag	'U' len L1	UID (L1)	HW tag	'H' len L2	HW (L2)	PLL tag	'P' len L3	PLL (L3)	Q tag	'Q' len L4	Q (L4)
Version info frame	!	V	x x x x	'U'	x x	L1 * x	'H'	x x	L2 * x	'P'	x x	L3 * x	'Q'	x x	L4 * x

ADC tag	'A' len L5	ADC (L5)	RFE tag	'F' len L6	RFE (L6)	SW tag	'S' len L7	SW (L7)	CP tag	'C' len L8	CP (L8)	Stop	
'A'	x x	L5 * x	'F'	x x	L6 * x	'S'	x x	L7 * x	'C'	x x	L8 * x	CR	LF

Figure 3: Version information frame format

Length field

Contains the length of the version frame excluding the start marker, identifier, the length field itself and the stop markers. Field size: 4 hex chars.

UID tag (‘U’)

Indicates the start of the UID field. Size: 1 hex char.

‘U’ len L1,

Contains the length of the UID field (number of chars). Field size: 2 hex chars.

UID (L1) field

The ‘Microcontroller UID’ field is a unique 24 byte unsigned HEX number, also see Table 3. Field size: variable.

HW tag (‘H’)

Indicates the start of the HW field. Size: 1 hex char.

‘H’ len L2

Contains the length of the HW field (number of chars). Field size: 2 hex chars.

HW (L2) field

Contains the baseboard hardware identifier, for example, ‘EA’ for the SiRad Easy or ‘SI’ for the SiRad Simple. Field size: variable.

PLL tag (‘P’)

Indicates the start of the PLL field. Size: 1 hex char.

‘P’ len L3

Contains the length of the PLL field (number of chars). Field size: 2 hex chars.

PLL (L3) field

Contains the PLL chip identifier, for example, ‘59’ for the ADF4159. Field size: variable.

CLK tag (‘Q’)

Indicates the start of the CLK field. Size: 1 hex char.

‘Q’ len L4

Contains the length of the CLK field (number of chars). Field size: 2 hex chars.

CLK (L4) field

Contains the CLK chip identifier. Field size: variable.

ADC tag (‘A’)

Indicates the start of the ADC field. Size: 1 hex char.

'A' len L5

Contains the length of the ADC field (number of chars). Field size: 2 hex chars.

ADC (L5) field

Contains the operating mode of the ADC, for example, 'I' for interleaved mode or 'N' non-interleaved mode. Field size: variable.

RFE tag ('F')

Indicates the start of the RFE field. Size: 1 hex char.

'F' len L6

Contains the length of the RFE field (number of chars). Field size: 2 hex chars.

RFE (L6) field

Contains the radar front end chip identifier, for example, '120_0x' for the 120 GHz chip, '024_0x' for the 24 GHz chip, or 'UIDENT' if the radar chip was not identified. Field size: variable.

Software version tag ('S')

Indicates the start of the software version field. Size: 1 hex char.

'S' len L7

Contains the length of the software version field (number of chars). Field size: 2 hex chars.

Software version (L7) field

Contains the software version as described below. Field size: variable.

<check-in ID >-<date >-<major>.<minor>.<revision>

Communication protocol version tag ('C')

Indicates the start of the protocol version field. Size: 1 hex char.

'C' len L8

Contains the length of the protocol version field (number of chars). Field size: 2 hex chars.

Protocol version (L8) field

Contains the protocol version as described below. Field size: variable.

<protocol ID>-<spec date>-<major>.<minor>.<revision>

2.3 System Information

The system info frame is used to uniquely identify different SiRad Easy® and return radar frontend information, see Figure 4.

System information	Start	Identifier	Microcontroller UID (24 Digits)						reserved	RFE MinFreq (5 Digits)	RFE MaxFreq (5 Digits)	Stop	Stop
System info frame	!	I	x	x	x	x	…	x	x x	x x x x x	x x x x x	CR	LF

Figure 4: System information frame format

After the start marker (1 byte) and the identifier ‘I’ (1 byte) follows the ‘UID’ field (24 bytes), which carries the UID of the microcontroller on the SiRad Easy®. Afterwards, there is a 2 byte reserved field then follow two 5 byte fields, which contain the minimum and maximum frequencies of the SiRad Easy® radar frontend. The frame ends with the stop markers ‘CR’ + ‘LF’.

Table 3: information frame format

Field	Ecoding	Example	Interpretation	Allowed values
Microcontroller UID	x - 24 byte string	‘800F0011570A 463332322039’	-	-
RFE MinFreq	x - 16 bit unsigned HEX between ‘00000’ and ‘FFFFF’	‘1D0D8’ -> 119000	0 to 119000 in MHz	‘00000’ to ‘FFFFF’
RFE MaxFreq	x - 16 bit unsigned HEX between ‘00000’ and ‘FFFFF’	‘1E848’ -> 125000	0 to 125000 in MHz	‘00000’ to ‘FFFFF’

Microcontroller UID field

The ‘Microcontroller UID’ field is a unique 24 byte unsigned HEX number (marked with ‘x’ in Figure 4), also see Table 33.

RFE MinFreq field

The ‘RFE MinFreq’ field is encoded as a 20 bit unsigned HEX number in 5 transmitted bytes (marked with ‘x’ in Figure 4). The data is recognized as ‘00000’ to ‘FFFFF’ characters in the terminal output, and are interpreted as values between 0 and 1048575 in MHz, also see Table 33. For example, ‘RFE MinFreq’ = 1D0D8 is interpreted as 0x1D0D8, which is 119000 in decimal range.

RFE MaxFreq field

The ‘RFE MaxFreq’ field is encoded as a 20 bit unsigned HEX number in 5 transmitted bytes (marked with ‘x’ in Figure 4). The data is recognized as ‘00000’ to ‘FFFFF’ characters in the terminal output, and are interpreted as values between 0 and 1048575 in MHz, also see Table 33. For example, ‘RFE MaxFreq’ = 1E848 is interpreted as 0x1E848, which is 125000 in decimal range.

2.4 Detailed Error Information

The error info frame includes error bits that may be raised during the signal processing of the radar data, see Figure 5.

Error information	Start	Identifier	Error flags (4 Digits)	Stop	Stop
Error info frame	!	E	X X X X	CR	LF

Figure 5: Error information frame format

The error info frame begins with the start marker (1 byte) and the identifier 'E' (1 byte) followed by the 'Error flags' field (4 byte), which is zero when no errors have been detected. The frame ends with the stop markers 'CR' + 'LF'.

Error flags field

The 'Error flags' field is transmitted as an 8 byte unsigned HEX number. Figure 6 shows the error bits in the 'Error flags' field. The error bits are explained below.

	Data								Processing								Basisband								Frontend							
Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ERROR_DETAILED	CRC	CRC	CRC	CRC	FLS	FLS	FLS	FLS	xxx	xxxx	xxxx	xxxx	ADC	ADC	ADC	ADC	xxxx	xxxx	xxxx	xxxx	PLL	PLL	PLL	PLL	RFE	RFE	RFE	RFE	RFE	RFE	RFE	RFE

Data	0	1	Processing	0	1	Basisband	0	1	Frontend	0	1
CRC	no error	reserved	ADC	no error	reserved	PLL	no error	reserved	RFE	no error	reserved
CRC	no error	reserved	ADC	no error	reserved	PLL	no error	Lock loss	RFE	no error	reserved
CRC	no error	reserved	ADC	no error	DC error	PLL	no error	Fmax not found	RFE	no error	RFE out of spec
CRC	no error	reserved	ADC	no error	Sample overrun	PLL	no error	Fmin not found	RFE	no error	VCO error
FLS	no error	reserved							RFE	no error	BW overrun
FLS	no error	reserved							RFE	no error	BW underrun
FLS	no error	reserved							RFE	no error	Fbase high
FLS	no error	reserved							RFE	no error	Fbase low

Figure 6: Detailed error flags

Error domains:

- CRC: <reserved>
- FLS: <reserved>
- XXX: <reserved>
- ADC: temporary ADC, sampling and data buffering errors
- XXX: <reserved>
- PLL: temporary PLL configuration errors, for example, operating range exceeded
- RFE: temporary radar frontend configuration errors, for example, operating range exceeded
-

Temporary errors are raised during processing but may go away when the parameter setting is changed. For example, when the parameters for the front end are manually changed so that its operating range is exceeded, a temporary RFE or PLL error may appear as long as this setting is applied.

2.5 Error Information

The error info frame includes error bits that may be raised during the signal processing of the radar data, see Figure 7. This frame will be send by default when the status update is enabled.

Error information	Start	Identifier	Error flags (4 Digits)	Stop	Stop
Error info frame	!	E	x x x x	CR	LF

Figure 7: Error information frame format

The error info frame begins with the start marker (1 byte) and the identifier ‘E’ (1 byte) followed by the ‘Error flags’ field (4 byte), which is zero when no errors have been detected. The frame ends with the stop markers ‘CR’ + ‘LF’.

Error flags field

The ‘Error flags’ field is transmitted as a 4 byte unsigned HEX number (marked with ‘x’ in Figure 7). Figure 8 shows the error bits in the ‘Error flags’ field. The error bits are explained below.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
ERROR	reserved								reserved								reserved		FLS	PRC	BB	PLL	RFE	CRC	reserved							

FLS	
0	no error
1	Flash error

BB	
0	no error
1	Baseband error

RFE	
0	no error
1	Frontend error

PRC	
0	no error
1	Processing error

PLL	
0	no error
1	PLL error

CRC	
0	no error
1	CRC error

Figure 8: Error flags

Error domains:

- FLS: <reserved>
- PRC: temporary errors in the signal processing
- BB: temporary baseband processing errors
- PLL: temporary PLL configuration errors
- RFE: temporary radar frontend configuration errors
- CRC: temporary errors in the UART transmission or CRC checksum

Temporary errors are raised during processing but may go away when the parameter setting is changed. For example, when the parameters for the front end are manually changed so that its operating range is exceeded, a temporary RFE or PLL error may appear as long as this setting is applied.

3 Commands

3.1 Frame Formats

Each command frame starts with ASCII value 33 ('!') as start marker and ends with two ASCII command characters ('CR' and 'LF') as stop marker.

Command frames

Configuration command	Start	dentifie	Command settings (8 Digits)	Stop	
System configuration	!	S	SYS_CONFIG	CR	LF
Radar frontend config	!	F	RFE_CONFIG	CR	LF
Baseband setup	!	B	BB_CONFIG	CR	LF

Special function comman	Start	dentifie	Stop	
Get full error report	!	E	CR	LF
Get system info	!	I	CR	LF
Do frequency scan	!	J	CR	LF
Trigger	!	M	CR	LF
Get version info	!	V	CR	LF

!	Start Marker, Identifier and Stop Marker	
x	Hex Digit	[0,1,2,...,A,B,C,D,E,F]
c	Ascii Character	[decimal 34 .. 255]

Figure 9: Command frames

The command frames are used to transmit configuration data to the SiRad Easy®. Different commands are used to configure several functionalities. Figure 9 shows the available command frames and Table4 lists their purpose. The blue parts in Figure 9 indicate start and stop markers, orange parts indicate data parts. The commands are further explained in the subsections below.

Table 4: Command frames overview, description

Command frame	Description
System configuration	Configure basic functions of the system
Radar frontend configuration	Configure frontend base-frequency and the VCO Divider
Baseband setup	Configure baseband and processing related parameters
Get system info	Request system info data
Do frequency scan	Request a scan of the frequency
Send Trigger	Send a trigger to start a measurement
Get full error info	Request detailed error data
Get version info	Request version info data

3.2 Send Commands

You can use a terminal program to send the command strings as, for example, with the Realterm terminal program. Calculate the command string by converting the command bits, an example is shown in Figure 10. Use zeros for any RESERVED fields.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SYS_CONFIG	SelfTrigDelay			reserved			LED		reserved							res	res	res	Gain		SER2	SER1	EXT	ST	res	res	res	res	res	res	SLF	PRE
DEFAULT EASY	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0
HEX	0				1				0				0				3				C				0				2			

Figure 10 : Example SYS_CONFIG command bits and hex format

Add the start marker '!' and the frame identifier (S, F, B) to the front of the hex command to form the command string. In case of the special function commands just use the start marker '!' and the command identifier (I, J, K, L, M, N) as the command string. In case of the example in Figure 10, you would get the command string

!S01003C02

In case of a special function command, it could look like this

!M

3.3 System Configuration

The system configuration command in Figure 11 is used to configure basic functions of the system, such as the triggering, frontend type, LED, data output interface, gain and data output modes.

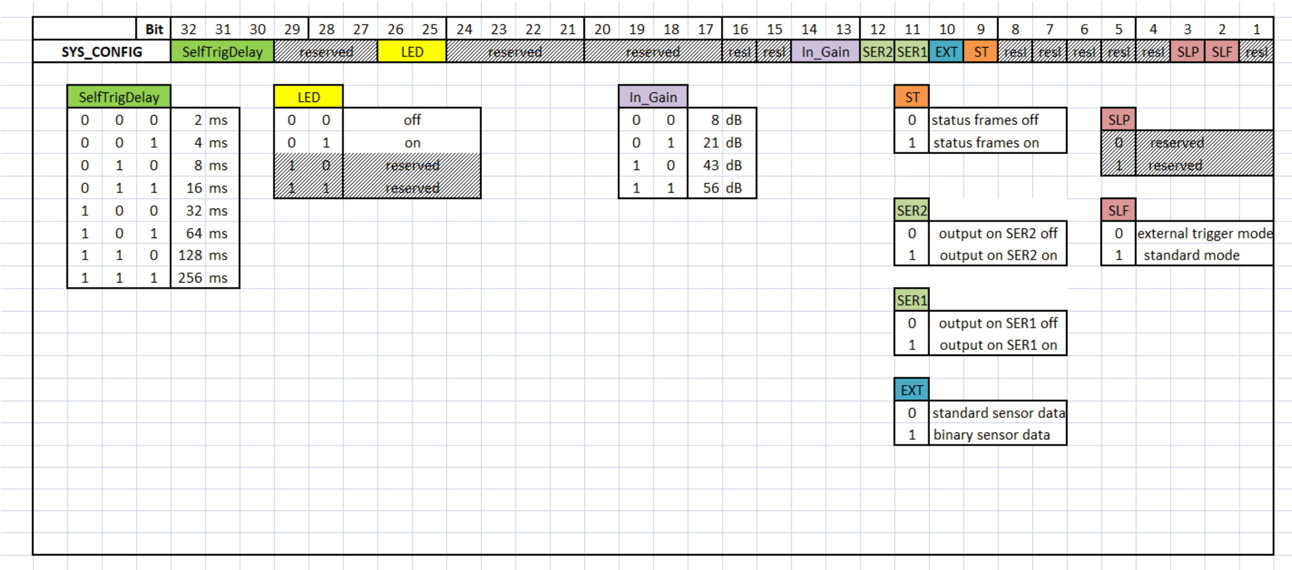

Figure 11: System configuration, SYS_CONFIG command frame

LED (2 bit) - LED operation

Selects the behavior of the onboard LED. The LED is switched off when disabled.

Gain (2 bit) - Manual Gain Control

You can set a suitable gain manually by choosing a setting from below for the 'Gain' Bits (13, 14) for standard The Sirad Easy®. Use one of the settings in Figure 12 for the 'Gain' Bits.

GAIN			
0	0	8	dB
0	1	21	dB
1	0	43	dB
1	1	56	dB

Figure 12: GAIN Bits in the SYS_CONFIG command

EXT (1 bit)

Select the data output mode. If this bit set to 0 Raw ADC data send in comma separated values decimal format as it is shown in Figure 1.

SER1 / SER2 (1 bit each) - Output Interface

Use these bits to choose the UART output interface of the SiRad Easy®. Configuration data can be fed to the SiRad Easy® using both UARTs at any time. Select SER1 for the WiFi connection of the SiRad Easy® and SER2 for the USB connection of the SiRad Easy®. Select SER1 for any output interface (UART-USB)

3.3.1 *Trigger Options*

SLF (1 bit) - Manual trigger or Self-trigger

When this bit is disabled, the system waits for an external trigger (Manual Trigger Mode). This is useful to minimize power consumption of the system when using longer measurement intervals.

When this bit is enabled, the SiRad Easy® triggers each measurement with an internal timer after 100 ms (Self-Trigger Mode). 'Pre-trigger' and 'Manual Trigger' are overridden with this bit.

SelfTrigDelay (3 bit) – Self-Trigger Delay

Sets a delay time between self-trigger events.

3.4 Radar Frontend Configuration

The radar frontend configuration command in Figure 13 is used to configure the radar frontend's base-frequency and the VCO Divider.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
RFE_CONFIG	VCO Divider (13 Bits, fixed per frontend)													Radar Frontend Base Frequency [MHz] (19 Bits)																		

VCO Divider (13 Bits, fixed per frontend)													
0	0	0	0	0	0	0	0	0	1	0	0	0	24 GHz
0	0	0	0	0	0	1	0	0	0	0	0	0	122 GHz
x	x	x	x	x	x	x	x	x	x	x	x	x	reserved

Radar Frontend Base Frequency [MHz] (19 Bits)																			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 MHz
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1 MHz
…																			…
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	524287 MHz

Figure 13: Radar frontend configuration, RFE_CONFIG command frame

VCO Divider (13 bit)

The VCO divider is a 13-bit unsigned integer value, so the theoretic value range is 0 to 8191. Please note that the VCO divider is fixed in hardware and frontend specific. The values for the 24 GHz and 122 GHz frontends are given in Figure 13.

RF Base-Frequency (19 bit)

The base-frequency is a 19-bit unsigned integer value interpreted in MHz, so the theoretic value range is 0 to 524287 MHz. Please note, that each frontend has a slightly different minimum and maximum operating frequency due to production tolerances. The frequencies supported by your frontend should be approximately in the range of 23300 to 26200 MHz for the 24 GHz frontend and 119100 to 125900 MHz for the 122 GHz frontend. The base-frequency should be chosen at least 100 MHz above the minimum operating frequency.

3.5 Baseband Setup

The baseband configuration command in Figure 14 is used to configure processing related parameters, like sampling parameters.

Bit	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
BB_CONFIG	reserved				resi	resi	Number of Samples														reserved			reserved			reserved			F_smp		

Number of Samples													
0	0	0	0	0	0	1	1	0	0	1	0	0	100
0	0	0	0	1	1	1	1	1	0	1	0	0	500
0	0	0	1	1	1	1	1	0	1	0	0	0	1000
0	0	1	1	1	1	1	0	1	0	0	0	0	2000

F_smp			
0	0	0	5.143
0	0	1	4.800
0	1	0	4.235
0	1	1	3.600
1	0	0	2.250
1	0	1	0.973
1	1	0	0.371
1	1	1	0.117

Figure 14: Baseband setup, BB_CONFIG command frame

3.5.1 *ADC Sampling Parameters*

#Samples - Number of Samples

'#Samples' is the number of samples taken. The number of samples is a 13 bit unsigned integer value. Maximum number of samples limited to 7500.

ADC ClkDiv - ADC clock divider / Sample frequency

'ADC ClkDiv' determines the ADC clock divider setting. 'ADC ClkDiv' is a 3 bit unsigned integer value. The value range is 0 to 7, according to the index of an internal look-up table which leads to the given number of MS/s according to the 'ADC ClkDiv' table in Figure 14.

Higher values result in longer A/D conversion times and can increase the signal strength of low signals.

3.6 Special Function Commands

Certain commands, explained in this section, use only a single letter to execute a function very fast. Send these commands three times in a row in case they are not executed.

Get fill error info - !E

Request a detailed error info frame at the next transmission slot

Get system info - !I

Request a system info frame at the next transmission slot.

Do frequency scan - !J

The system scans the maximum and minimum usable frequency of the installed frontend at every startup. To trigger that scan manually at runtime, use this command.

Send Trigger - !M

Software command to trigger a measurement.

Get version info - !V

Request a version info frame at the next transmission slot.

4 Extended Data

Figure 15 shows supported extended data frames and raw data of the AD converter is transmitted. The blue part indicates header and stop markers. The purple areas indicate channel number, data length and the transmitted raw ADC data.

Binary output

Raw sensor data	Header	Chn	Length	Data (Length * 2 Bytes)	Stop	
reserved	$AA $CA $FE	0	Size n	uint16	CR	LF
ADC frame interleaved		1	uint16	uint16		

Figure 15: Extended binary mode frame

When the Extended mode is selected from the System configuration, both of status and error frame are send in same binary format given in Figure 15. Figure 16 shows terminal output of the binary mode.

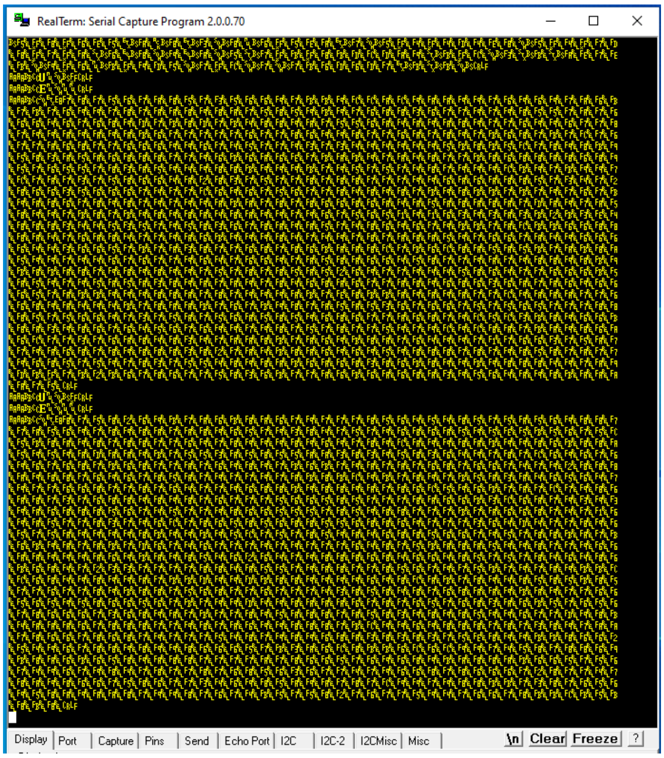

Figure 16: Extended mode in binary format

5 Disclaimer / License

By using the information in this document, our products and the provided software, you agree to our disclaimer and license conditions.

| Disclaimer |

==========

The information contained in this document is subject to change at any time without notice.

Silicon Radar GmbH assumes no responsibility or liability for any loss, damage or defect of a product that is caused in whole or in part by

(i) use of any circuitry other than circuitry embodied in a Silicon Radar GmbH product,

(ii) misuse or abuse including static discharge, neglect or accident,

(iii) unauthorized modification or repairs which have been soldered or altered during assembly and are not capable of being tested by Silicon Radar GmbH under its normal test conditions, or

(iv) improper installation, storage, handling, warehousing or transportation, or

(v) being subjected to unusual physical, thermal, or electrical stress.

Silicon Radar GmbH makes no warranty of any kind, express or implied, with regard to this material, and specifically disclaims any and all express or implied warranties, either in fact or by operation of law, statutory or otherwise, including the implied warranties of merchantability and fitness for use or a particular purpose, and any implied warranty arising from course of dealing or usage of trade, as well as any common-law duties relating to accuracy or lack of negligence, with respect to this material, any Silicon Radar product and any product documentation. Products sold by Silicon Radar are not suitable or intended to be used in a life support applications or components to operate nuclear facilities, or in other mission critical applications where human life may be involved or at stake. All sales are made conditioned upon compliance with the critical uses policy set forth below.

CRITICAL USE EXCLUSION POLICY: BUYER AGREES NOT TO USE SILICON RADAR GMBH'S PRODUCTS FOR ANY APPLICATION OR IN ANY COMPONENTS USED IN LIFE SUPPORT DEVICES OR TO OPERATE NUCLEAR FACILITIES OR FOR USE IN OTHER MISSION-CRITICAL APPLICATIONS OR COMPONENTS WHERE HUMAN LIFE OR PROPERTY MAY BE AT STAKE.

Silicon Radar GmbH owns all rights, title and interest to the intellectual property related to Silicon Radar GmbH's products, including any software, firmware, copyright, patent, or trademark. The sale of Silicon Radar GmbH products does not convey or imply any license under patent or other rights. Silicon Radar GmbH retains the copyright and trademark rights in all documents, catalogs and plans supplied pursuant to or ancillary to the sale of products or services by Silicon Radar GmbH. Unless otherwise agreed to in writing by Silicon Radar GmbH, any reproduction, modification, translation, compilation, or representation of this material shall be strictly prohibited.

Silicon Radar GmbH, 2019

| License |

========

The radar software delivered with the Install Package is provided by Silicon Radar GmbH.

The software is provided ‘as is’ for testing purposes only. While hoping it will be useful, there is NO WARRANTY of any kind, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. In no event shall the distributor, the authors, any of the contributors, or any third party be liable for any direct, indirect, incidental, or consequential damages (including, but not limited to, procurement of substitute goods and services, loss of use, damage to hardware, loss of data, loss of profits, or business interruption).

All software in the Install Package marked as third-party is copyrighted by their respective owners and covered by their respective licenses. Please see the corresponding websites or the package information for details. Please also see the README.txt files of the Install Package for details.

Silicon Radar GmbH, 2019